



ADSL2+ x86 Router

User Manual – Linux

Revision 0.4

Table of Contents

1. Introduction.....	5
Key Features.....	5
2. HARDWARE	6
2.1 Front Panel Indicators.....	6
2.2 Rear Panel Connections.....	7
2.3 External LEDs and Switch Connections.....	8
2.4 ADSL Filter Connections.....	9
3. Linux ADSL Setup	10
3.1 Linux Driver.....	10
3.2 Kernel Configuration.....	10
3.3 ATM utilities and libraries.....	16
3.4 Bridging (RFC1483 / RFC2684) Utility for Bridged Ethernet.....	16
3.5 PPPoA Packages.....	16
3.6 PPPoE Packages.....	16
4. Linux ADSL Showtime.....	17
4.1 Bridged Ethernet (RFC1483).....	17
4.2 PPPoE.....	17
4.3 PPPoA (RFC2364).....	17
5. ML-PPP.....	18
5.1 PPPoE Setup for ML-PPP.....	18
5.2 PPPoA Setup for ML-PPP.....	19
5.3 ML-PPP Link Monitoring.....	20
6. Ethernet Setup.....	23
Appendix A - ADSL Protocol and PVC Values.....	24
Appendix B – CLI and ADSL Parameters.....	25
Appendix C – Firmware upgrades.....	28
C.1 ADSL Firmware Flash Upgrade.....	28
APPENDIX D – ADSL Troubleshooting.....	29
D.1 Loading Modules.....	29
D.2 Line Sync.....	29
D.3 TX and RX packets.....	29
D.4 LCP timeouts (PPPoE and PPPoA only).....	29
D.5 Authentication Failure (PPPoE and PPPoA only).....	30
D.6 ADSL Troubleshooting.....	30
APPENDIX E – Programmable LEDs & Switches.....	33
Copyright.....	34

This page is intentionally blank

SAFETY WARNINGS



CAUTION

(i) Risk of explosion if the battery is replaced with an incorrect type. Batteries should be recycled where possible. Disposal of used batteries must be in accordance with local environmental regulations.

(ii) The Router PCB assembly should be installed in a CE compliant Chassis/Case that is fully enclosed and can only be opened with the use of a tool or key

(iii) Do not connect the ADSL ports on the Router to the Telecommunications Network until the PCB assembly is enclosed in a suitable Chassis/Case.

(iv) In the event that the Router PCB assembly has to be removed, the ADSL ports on the Router must be disconnected from the Telecommunications Network before opening. Do not re-connect the ADSL ports until the Router PCB assembly is case is re-installed in an enclosed case.

WARNING

This is a class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

1. Introduction

Geos is a family of x86 Router boards with integrated ADSL2+ ports. Designed with Linux in mind, these boards run standard x86 Linux. Unlike any other ADSL router on the market, there is no need for developers to port code to run on other architectures such as MIPS or ARM. The same code used on a PC platform, will run on a Geos. This radically reduces the software effort to build a Linux appliance.

These boards are suitable for a wide range of applications including...

- Firewalls and gateways
- VOIP and PBX appliances
- Internet appliances
- Wireless hotspots
- Bonding routers (*3rd party bonding software required*)
- ML-PPP routers

Key Features

- 500MHz AMD LX800 CPU (x86 Instruction set)
- Single or dual integrated ADSL2+ ports
- Annex A / M with up to 2.5Mbps upstream per port
- Annex B option available on request
- Dual 10/100 Ethernet interfaces
- Serial console port
- USB via rear panel
- Mini-PCI expansion slot for Wireless & VOIP applications
- 12VDC power
- Compact Flash slot for diskless operation
- 44Pin IDE header for optional 2.5" HDD
- All device drivers are part of the current 2.6 Linux kernel

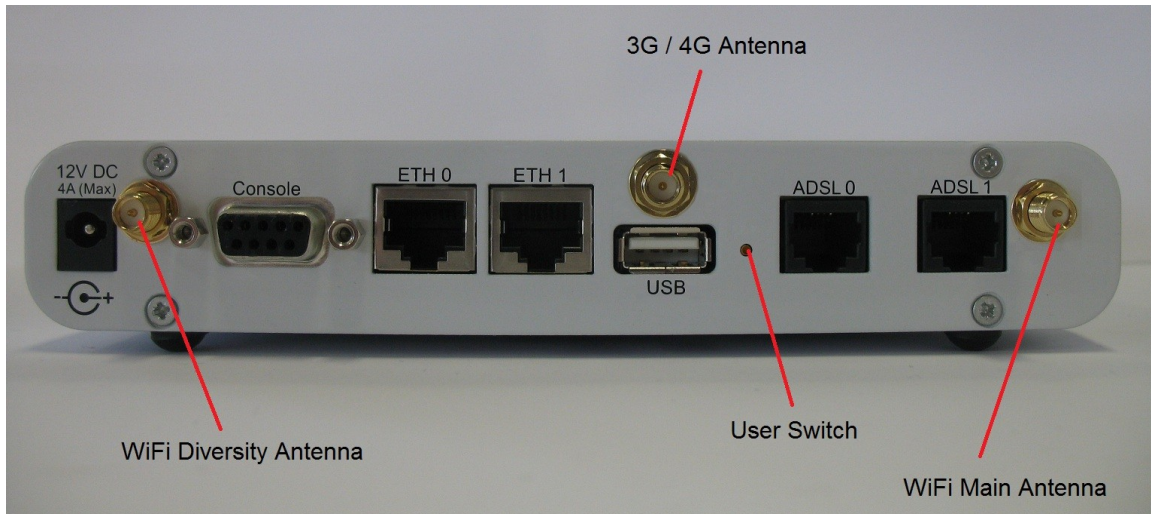
2. HARDWARE

2.1 Front Panel Indicators



Label	Colour	Function
PWR	Green	DC Power ON Indicator
TX/RX ADSL0	Yellow	Flashes with TX or RX data, ADSL0
LINK ADSL0	Green	Flashing indicates ADSL0 training On indicates ADSL0 Line Sync
TX/RX ADSL1	Yellow	Flashes with TX or RX data, ADSL1
LINK ADSL1	Green	Flashing indicates ADSL1 training On indicates ADSL1 Line Sync
TX/RX ETH0	Yellow	Flashes with TX or RX data, ETH0
LINK ETH0	Green	On indicates ETH0 Link up
TX/RX ETH1	Yellow	Flashes with TX or RX data, ETH1
LINK ETH1	Green	On indicates ETH1 Link up
LED0	Yellow	Software controlled LEDs, see Appendix E for details
LED1	Green	
LED2	Blue	
IDE	Red	Flashed with IDE activity

2.2 Rear Panel Connections



12VDC : DC Power Input

9 – 14 VDC

5.5/2.5mm connector, centre positive

Typical current 1.0A (12W), Max current 3.0A

Console: Serial Port

RS-232

115,200 bps

8 Data bits, 1 Stop bit, No parity

ETH0 / ETH1 : Ethernet ports

10/100 Mbps

USB : USB port

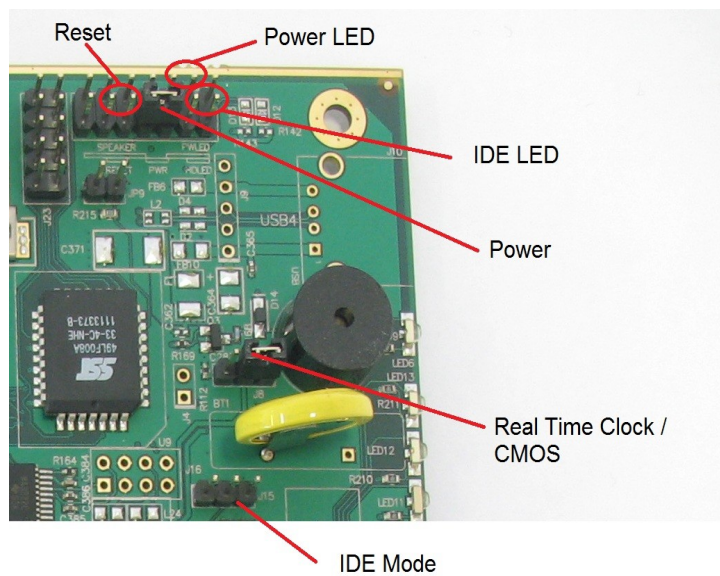
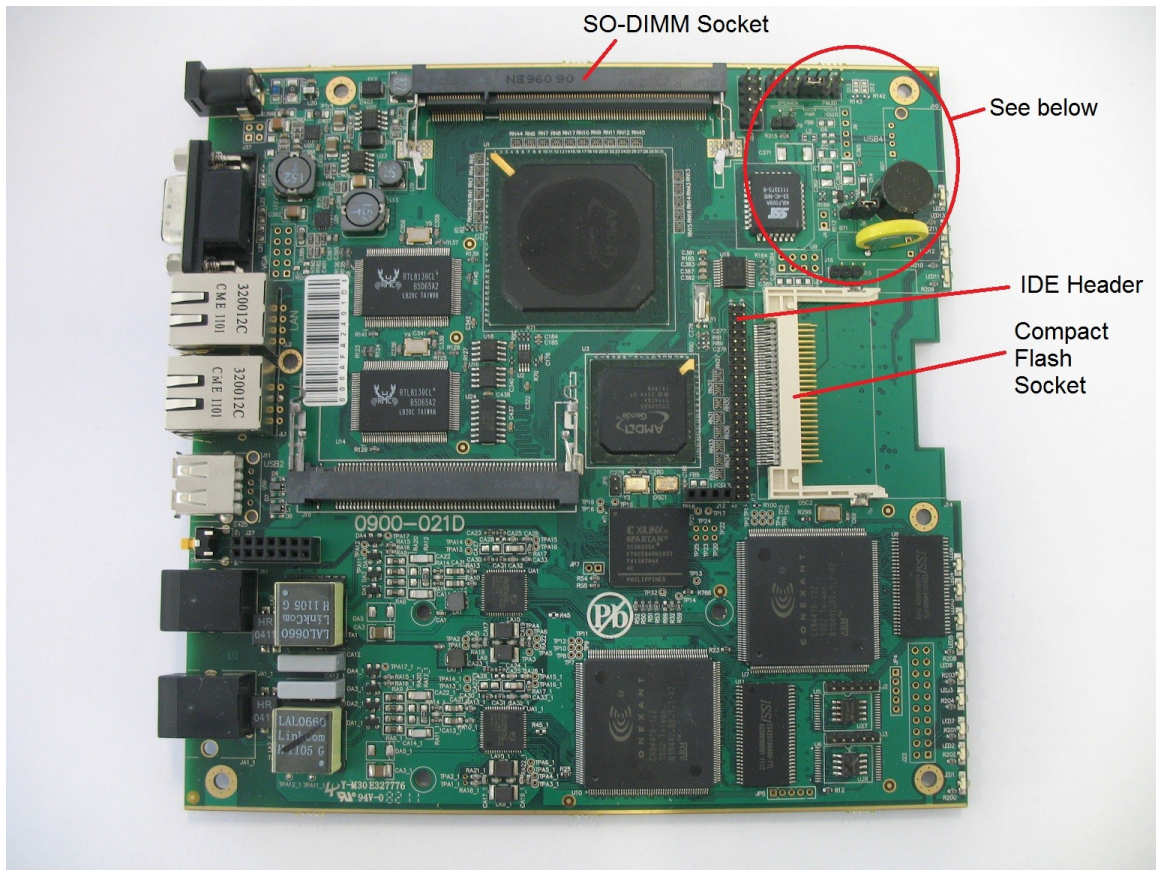
Supports USB1 & USB2

User Switch : General purpose user input, push button

ADSL0 / ADSL1 : ADSL Ports

Supports ADSL1, ADSL2, ADSL2+ Annex A/M

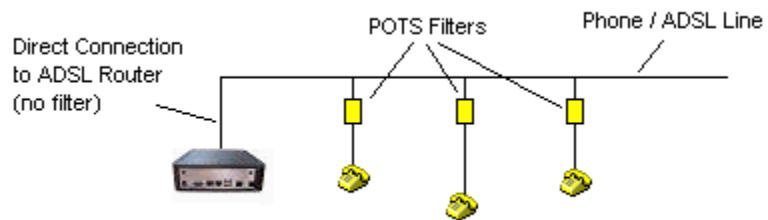
2.3 External LEDs and Switch Connections



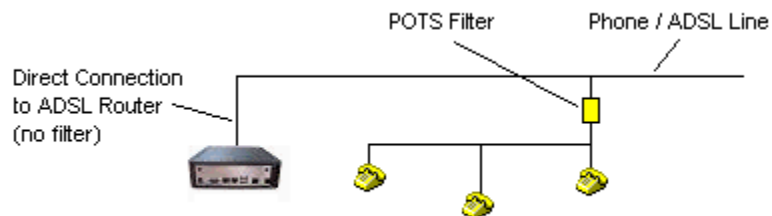
2.4 ADSL Filter Connections

If your ADSL connection is also used by POTS (Plain Old Telephone System) devices such as phones, fax machines and answering machines, they must be filtered to prevent them from interfering with your ADSL connection.

The diagram below shows how POTS filters can be connected



It is also possible for several POTS devices to share the same filter



Important : DO NOT place a POTS filter in line between your ADSL Router and the Phone / ADSL line.

3. Linux ADSL Setup

3.1 Linux Driver

The ADSL hardware on the Geos uses the GPL Linux Solos driver. Linux kernels 2.6.23 and later are currently supported. The following table indicates which driver to use for various kernels :

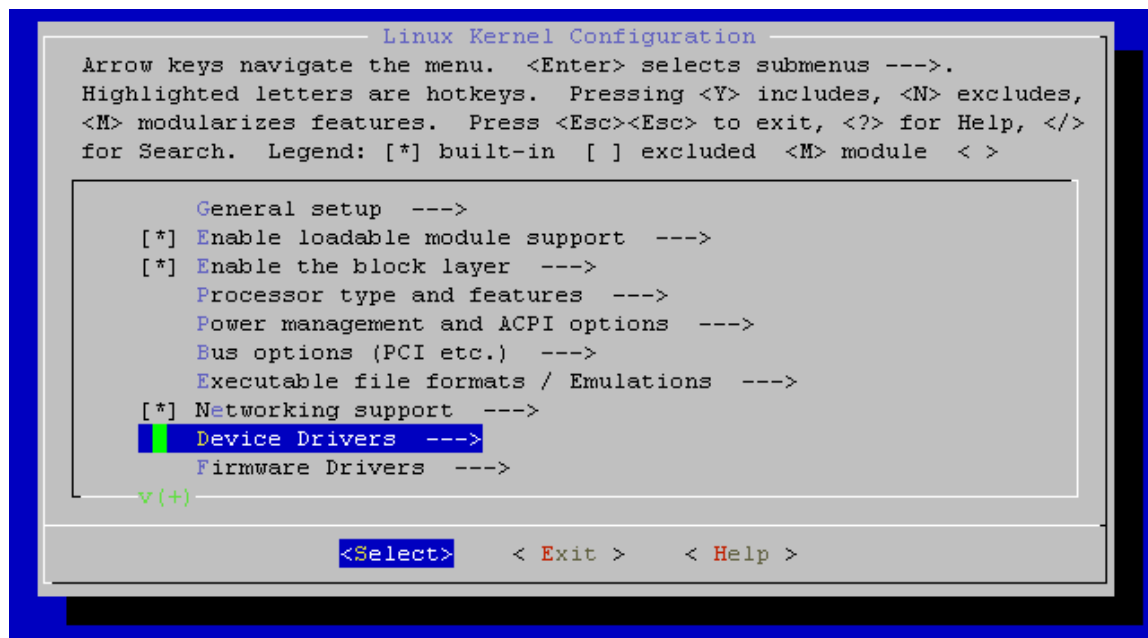
Kernel	Source	Comments
2.6.23 – 2.6.28	http://sourceforge.net/projects/openadsl/files/linux-solos-driver/	Not part of these standard kernels.
2.6.29, 2.6.30	http://sourceforge.net/projects/openadsl/files/linux-solos-driver/	Apply patches to standard kernel.
2.6.31 and later	Included in standard kernel source	
3.0.1 and later	Included in standard kernel source	

3.2 Kernel Configuration

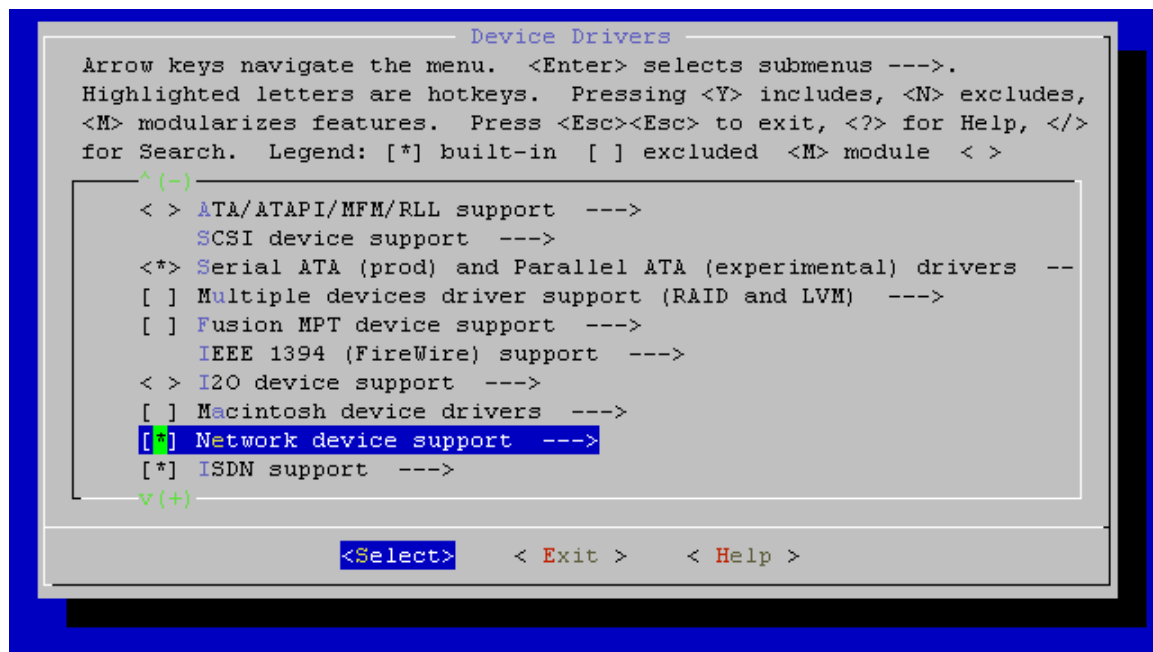
This manual assumes the user is familiar with building 2.6 Linux kernels. If you haven't built a kernel before please read the Kernel Rebuild Guide at...

<http://www.digitalhermit.com/linux/Kernel-Build-HOWTO.html>

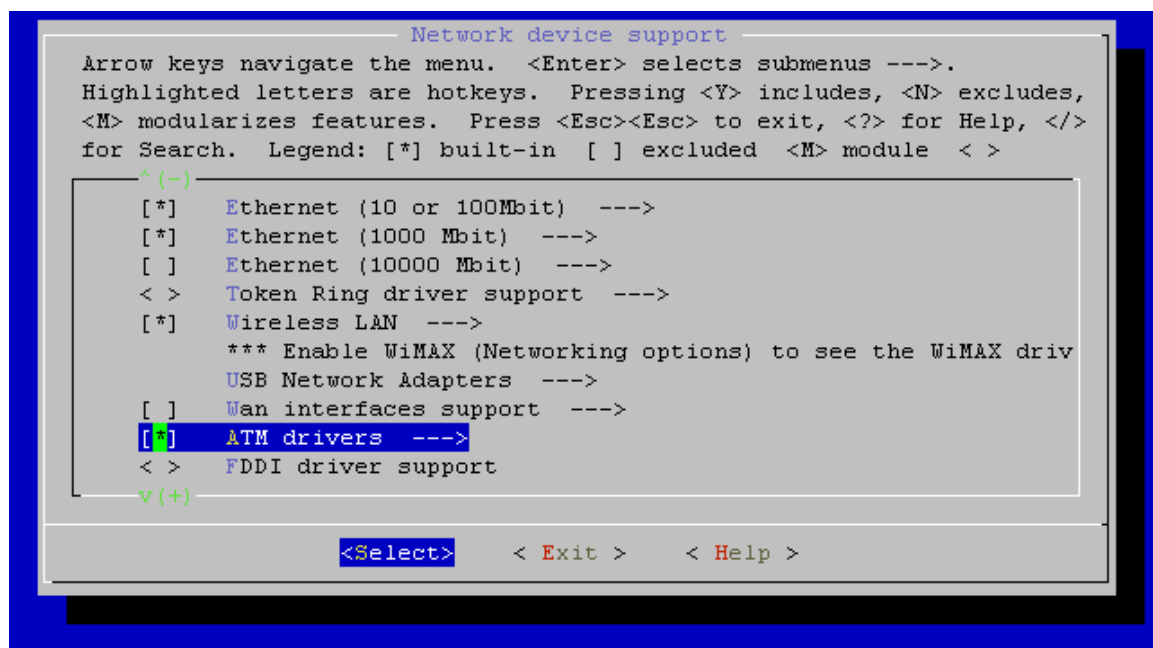
From your kernel source directory run **make menuconfig** and select Device Drivers...



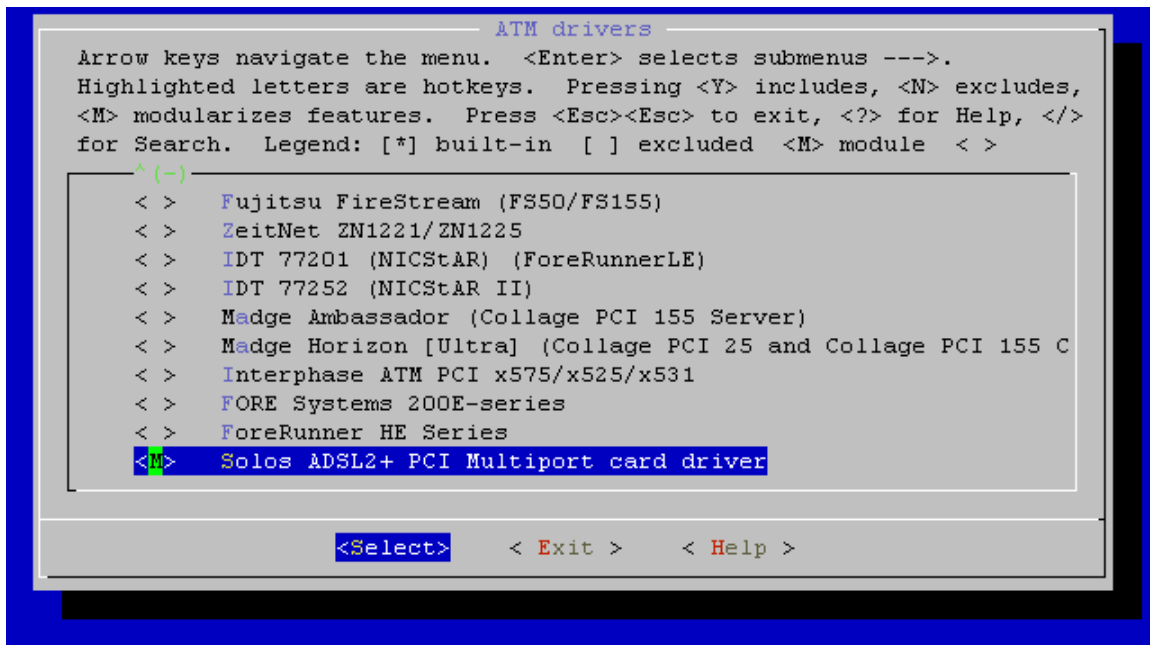
Scroll down to Network device support and select it ...



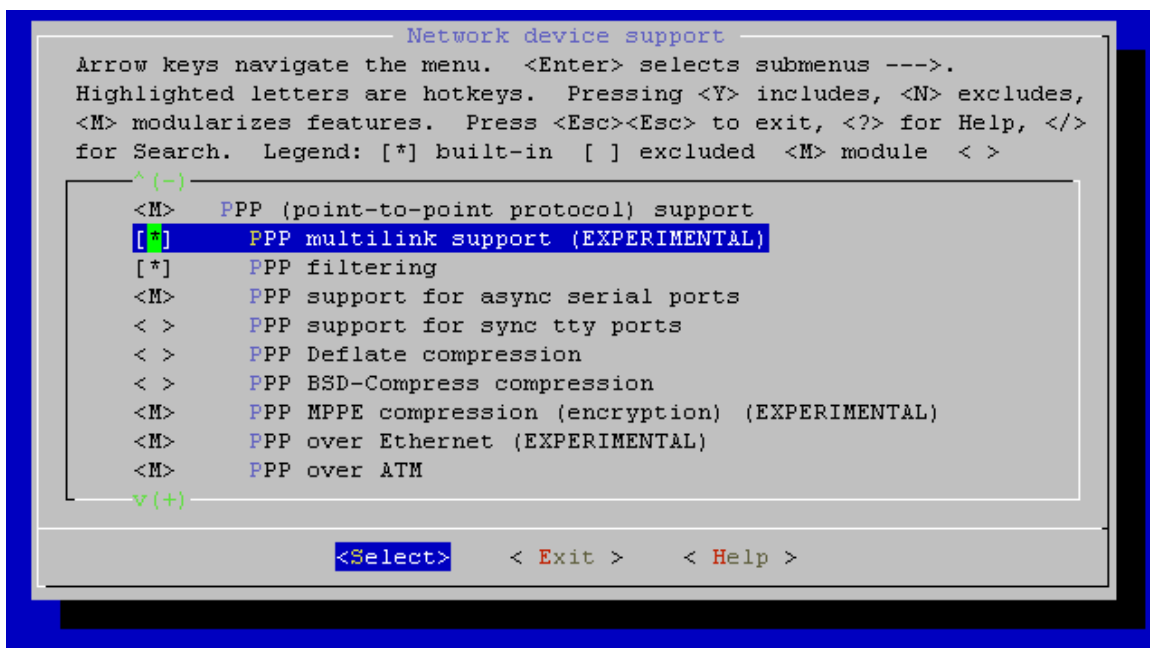
Scroll down to ATM drivers and select it ...



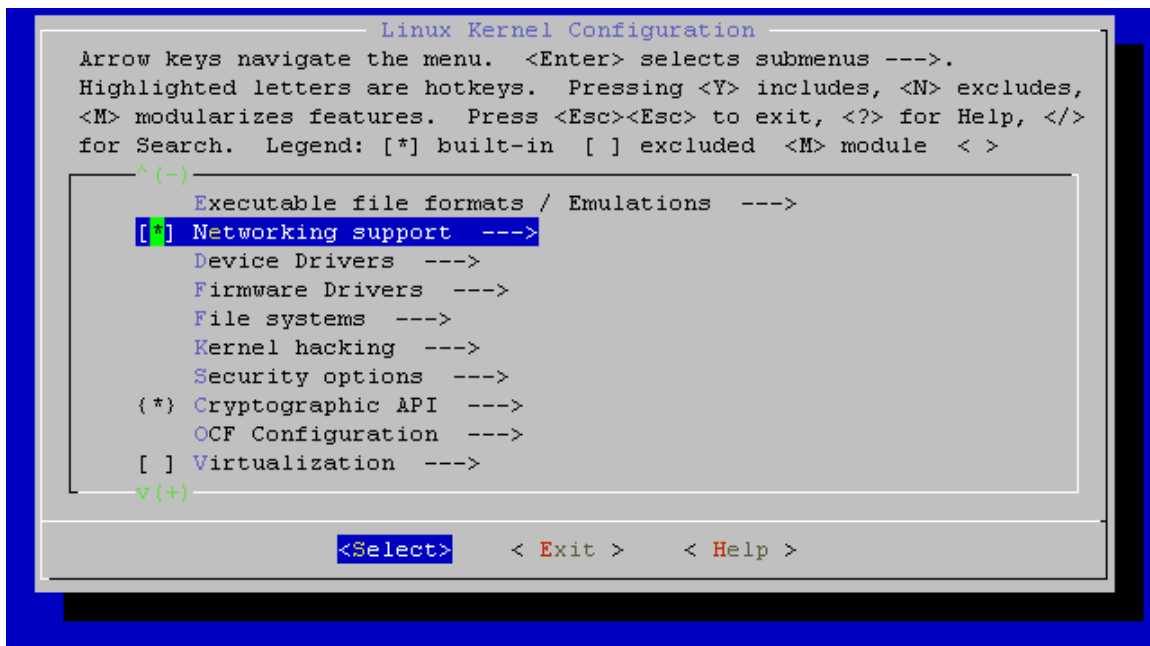
Configure the Solos driver as a module ...



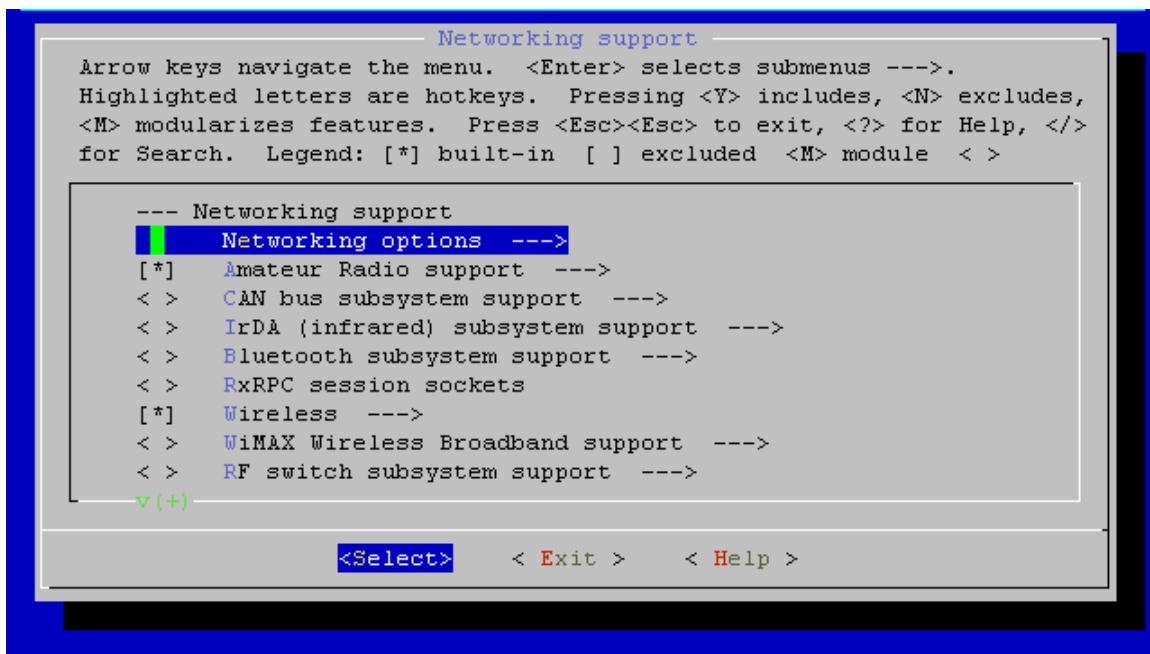
Then select Exit to return to the Network device support menu.
Scroll down to the PPP options and configure as shown ...



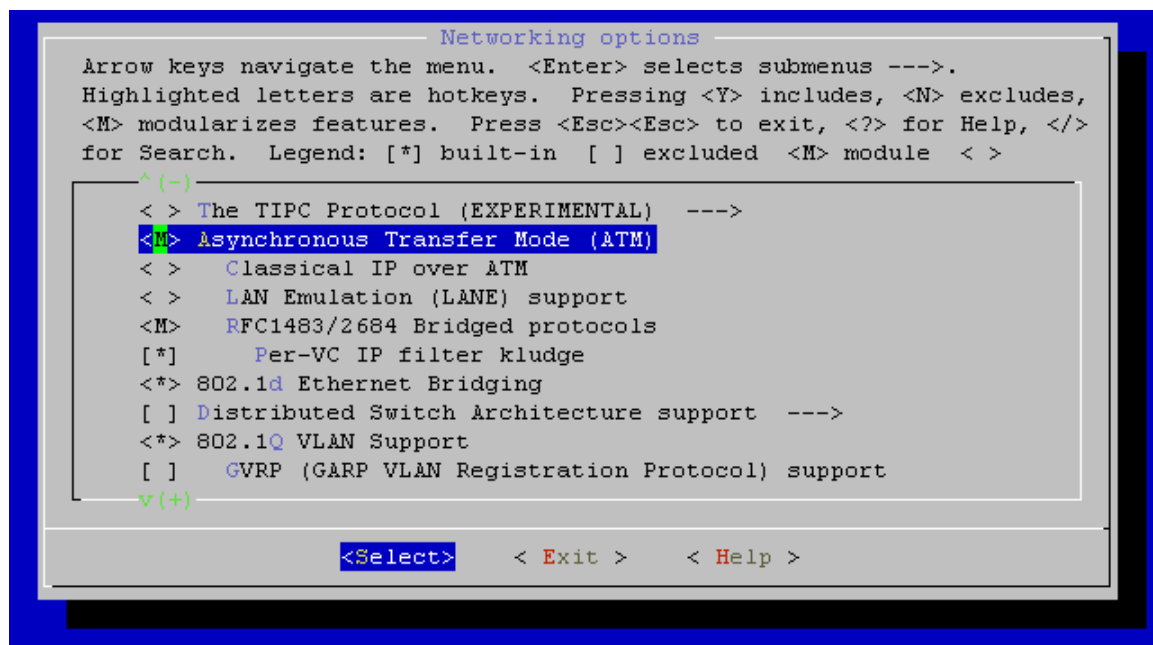
Select Exit to return to the Kernel Config. Menu.
Scroll up and select Networking support ...



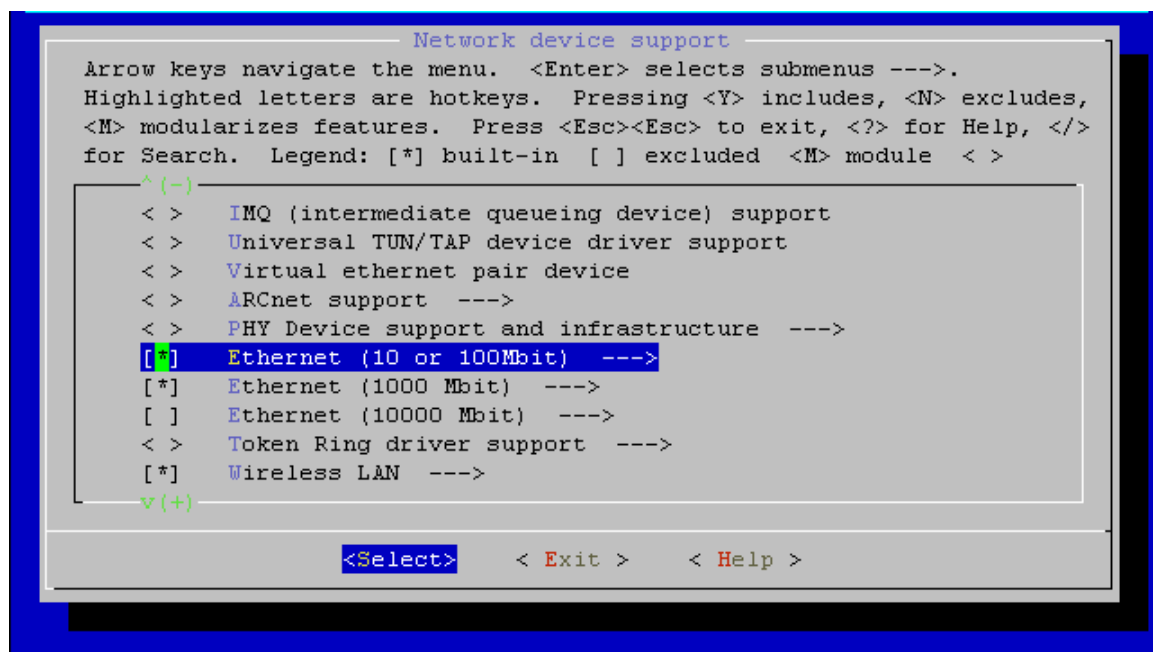
Next scroll down to Networking options and select it ...



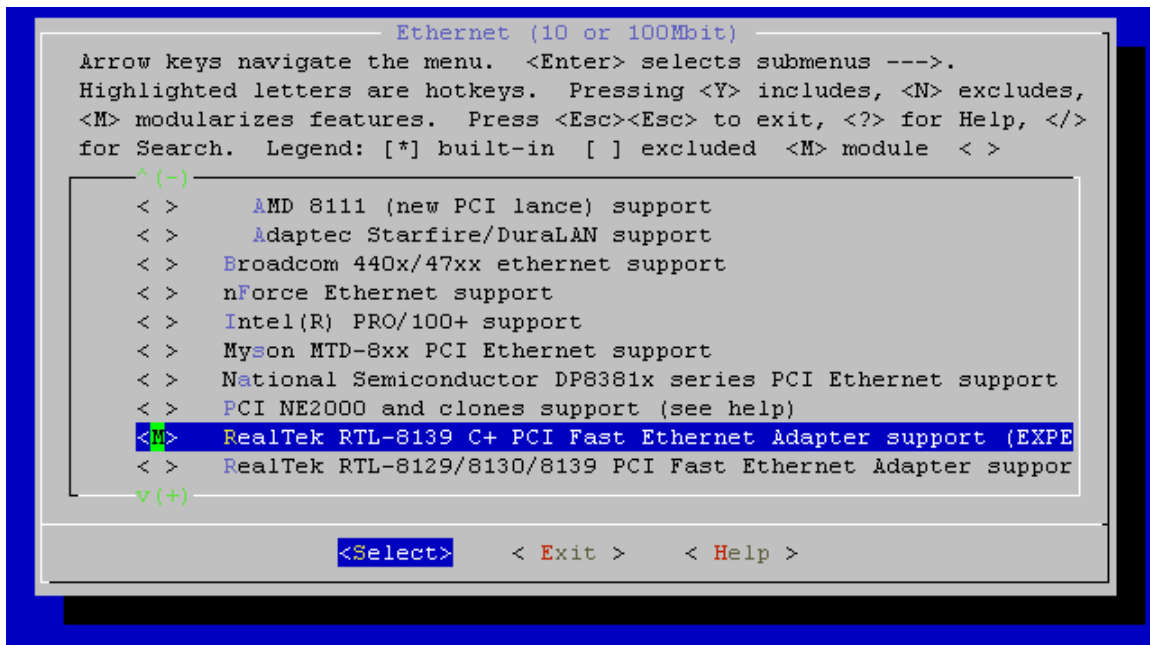
Scroll down to the ATM options and configure as shown ...



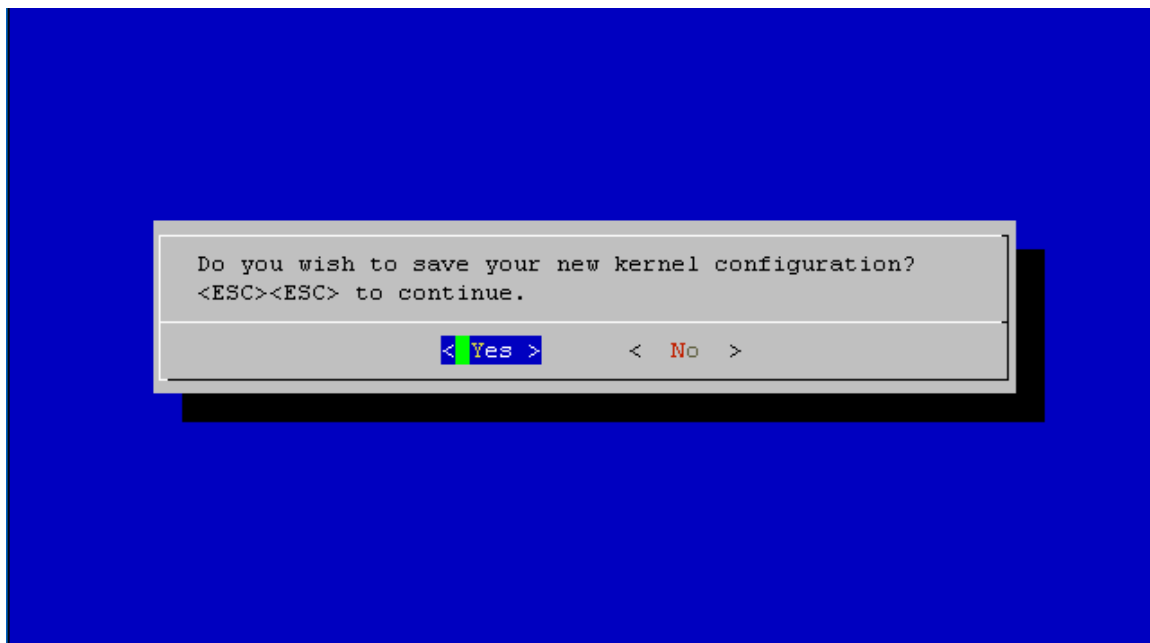
The select Exit and return to the Network device support menu.
Scroll down to the Ethernet (10/100) drivers and select ...



Scroll down to RTL-8139C+ and configure as a module...



Finally exit all menus and you will be prompted to save the new config and Select Yes...



Now your kernel is configured. Your kernel & modules will need to be recompiled and installed. If you are not sure how to do this please refer the Kernel Rebuild Guide at...

<http://www.digitalhermit.com/linux/Kernel-Build-HOWTO.html>

3.3 ATM utilities and libraries

Fedora : install the linux-atm package

Debian : install the libatm1-dev package

Source : linux-atm-2.5.0.tar.gz from <http://sourceforge.net/projects/linux-atm>

3.4 Bridging (RFC1483 / RFC2684) Utility for Bridged Ethernet

If you plan to use RFC1483 (Bridged Ethernet) you will need br2684ctl.

Fedora : supplied with the linux-atm package

Debian : install the br2684ctl package

Source : supplied with linux-atm-2.5.0.tar.gz

3.5 PPPoA Packages

Fedora : install the ppp package

Debian : install the ppp package

Source : ppp-2.4.4.tar.gz from <ftp://ftp.samba.org/pub/ppp/>

3.6 PPPoE Packages

Download rp-pppoe-3.10.tar.gz from <http://www.roaringpenguin.com/products/pppoe>

To Install

```
$ tar -xvzf rp-pppoe-3.10.tar.gz  
$ cd rp-pppoe-3.10  
$ ./configure --enable-plugin=$PPPPDIR  
$ make  
$ make install
```

Note \$PPPPDIR is the ppp directory on your system, eg. /usr/local/lib/pppd/2.4.4

4. Linux ADSL Showtime

IMPORTANT : The following examples use Vpi = 0, Vci = 38 which are the common PVC values for the UK. Other countries & providers are different, see Appendix A or check with your provider first.

4.1 Bridged Ethernet (RFC1483)

```
$ modprobe atm
$ insmod solos-pci.ko
$ br2684ctl -b -s 2040 -c 0 -a 0.0.38
$ ifconfig nas0 203.123.69.1 up
```

and for the second port...

```
$ br2684ctl -b -s 2040 -c 1 -a 1.0.38
$ ifconfig nas1 203.123.69.2 up
```

4.2 PPPoE

```
$ modprobe atm
$ insmod solos-pci.ko
$ br2684ctl -b -s 2040 -c 0 -a 0.0.38
$ ifconfig nas0 up
$ pppd plugin rp-pppoe.so nic-nas0 user test password test noauth
```

and for the second port...

```
$ br2684ctl -b -s 2040 -c 1 -a 1.0.38
$ ifconfig nas1 up
$ pppd plugin rp-pppoe.so nic-nas1 user test2 password test2 noauth
```

4.3 PPPoA (RFC2364)

```
$ modprobe atm
$ modprobe pppoatm
$ insmod solos-pci.ko
$ pppd plugin pppoatm.so 0.0.38 user test password test noauth
```

and for the second port...

```
$ pppd plugin pppoatm.so 1.0.38 user test2 password test2 noauth
```

5. ML-PPP

If your provider supports ML-PPP, you can bond both ADSL ports together and obtain close to double the speed of a single port.

5.1 PPPoE Setup for ML-PPP

To set up PPPoE you need to create two ethernet interfaces first using br2684ctl:

USAGE: br2684ctl -b -s 2040 -c [port num] -a [port].[VPI].[VCI]

br2684ctl -b -s 2040 -c 0 -a 0.8.35

br2684ctl -b -s 2040 -c 1 -a 1.8.35

ifconfig nas0 up

ifconfig nas1 up

IMPORTANT : The following above examples uses Vpi = 8, Vci = 35 which are the common PVC values for Australia and France. Other countries & providers are different, see Appendix A or check with your provider first.

This will create two interfaces nas0 and nas1.

Then create two files in /etc/ppp/peers called dsl0 and dsl1 as shown below.

The file **dsl0** should contain the following :

plugin rp-pppoe.so nic-nas0
user yourusername@yourprovider
noauth
defaultroute
usepeerdns
mp

The file **dsl1** should contain the following:

plugin rp-pppoe.so nic-nas1
user yourusername@yourprovider
noauth
defaultroute
usepeerdns
mp

Then try:

pppd call dsl0

Wait ~10 seconds for the first PPP session to come up, then:

pppd call dsl1

5.2 PPPoA Setup for ML-PPP

IMPORTANT : The following above examples uses Vpi = 8, Vci = 35 which are the common PVC values for Australia and France. Other countries & providers are different, see Appendix A or check with your provider first.

This will create two interfaces nas0 and nas1.

Then create two files in /etc/ppp/peers called dsl0 and dsl1 as shown below.

The file **dsl0** should contain the following :

```
pppd plugin pppoatm.so 0.8.35  
user yourusername@yourprovider  
noauth  
defaultroute  
usepeerdns  
mp
```

The file **dsl1** should contain the following:

```
pppd plugin pppoatm.so 1.8.35  
user yourusername@yourprovider  
noauth  
defaultroute  
usepeerdns  
mp
```

Then try:

```
pppd call dsl0
```

Wait ~10 seconds for the first PPP session to come up, then:

```
pppd call dsl1
```

5.3 ML-PPP Link Monitoring

For a permanent ML-PPP connection, a monitoring script (daemon) must be used to handle the error conditions that pppd can not. Otherwise users will experience excessive delays and freezes in the case where one link goes down or there is a line fault.

Important : *The following example and script is written for PPPoE.*

First create two files in /etc/ppp/peers called dsl0 and dsl1 as shown below.
The file **dsl0** should contain the following :

```
plugin rp-pppoe.so nic-nas0  
user yourusername@yourprovider  
noauth  
defaultroute  
usepeerdns  
mp
```

The file **dsl1** should contain the following:

```
plugin rp-pppoe.so nic-nas1  
user yourusername@yourprovider  
noauth  
defaultroute  
usepeerdns  
mp
```

Second create a script called mlppp.sh as shown overleaf.

TIPS

- (I) Don't forget to configure the VPI and VCI for your provider – see Appendix A
- (ii) Change the file attributes so that mlppp.sh is executable
- (iii) Run it !

mlppp.sh

```
#!/bin/bash

VPI=8
VCI=35

#Timeout for pppd to come up
TIMEOUT=5
ADSL_STATE[0]=0
ADSL_STATE[1]=0
MAGIC=""

function gen_magic() {
    A=$((RANDOM % 256))
    B=$((RANDOM % 256))
    C=$((RANDOM % 256))
    D=$((RANDOM % 256))
    MAGIC=`printf "%02x%02x%02x%02x" $A $B $C $D`
}

function check_adsl() {
    RESTART_PPP=0
    for port in 0 1 ; do
        A=`cat /sys/class/atm/solos-pci${port}/parameters/Connected`
        if [ "${A}" == "true" ] ; then
            if [ ${ADSL_STATE[$port]} -eq 0 ] ; then
                #echo "ADSL port ${port} has line sync";

                #don't change last magic number
                start_adsl "$port"
            fi
            ADSL_STATE[$port]=1;
        else
            if [ "${A}" == "false" ] ; then
                if [ ${ADSL_STATE[$port]} -eq 1 ] ; then
                    #echo "ADSL port ${port} lost line sync"
                    RESTART_PPP=1
                fi
                ADSL_STATE[$port]=0;
            fi;
        fi;
    done
    if [ $RESTART_PPP -eq 1 ] ; then
        kill_pppd
        gen_magic
        start_adsl
    fi
}

function check_pppd() {
    PROCESSES=`ps -C pppd | grep "pppd" | wc -l`
    LINKSUP=0
    for port in 0 1 ; do
        A=`cat /sys/class/atm/solos-pci${port}/parameters/Connected`
        if [ "${A}" == "true" ] ; then
            LINKSUP=$((LINKSUP + 1));
        fi;
    done
    if [ $LINKSUP -ne $PROCESSES ] ; then
        #echo "$LINKSUP links != $PROCESSES processes"
        kill_pppd
        gen_magic
        start_adsl
    # else
    #     check_ping;
    # fi
}

#kill pppd processes
function kill_pppd() {
    echo "Killing PPPD"
    if [ -e "/var/run/ppp0.pid" ] ; then
```

```

        pids=`ps -C pppd | grep "pppd" | awk '{print $1}'`
        kill -s TERM ${pids};
    ps -C pppd | grep -q "pppd"
    while [ $? -eq 0 ] ; do
        sleep 1;
        ps -C pppd | grep -q "pppd"
    done
fi
}

#function check_ping() {
#    ping -I ppp0 -c 1 -w 1 200.200.200.1 &> /dev/null
#    if [ $? -ne 0 ] ; then
#        echo "Ping check failed"
#        kill_pppd;
#        gen_magic
#        start_adsl;
#    fi
#}

function start_adsl() {
    if [ "$1" == "" ] ; then
        portlist="0 1";
    else
        portlist=$1;
    fi
    for port in $portlist ; do
        A=`cat /sys/class/atm/solos-pci${port}/parameters/Connected`
        if [ "${A}" == "true" ] ; then
            ADSL_STATE[${port}]=1
            echo "Starting PPPD on port ${port}"
            pppd call dsl${port} endpoint magic:${MAGIC} &> /dev/null
        else
            ADSL_STATE[${port}]=0
            #echo "ADSL port ${port} doesn't have line sync"
            if [ "${A}" != "false" ] ; then
                echo "There may be a problem with ADSL port ${port}";
            fi;
        fi;
        sleep $TIMEOUT
    done
}

kill_pppd

#kill existing nas interfaces
for port in 0 1 ; do
    if [ ! -e "/var/run/nas${port}.pid" ] ; then
        br2684ctl -b -s 2040 -c ${port} -a ${port}.${VPI}.${VCI}
        ifconfig nas${port} hw ether "00:0A:FA:00:00:0${port}" up;
    fi;
done

gen_magic
start_adsl

while true ; do
    check_adsl;
    check_pppd;
    sleep 3;
done

```

6. Ethernet Setup

```
modprobe 8139cp
```

```
ifconfig eth0 192.168.0.1 up
```

```
ifconfig eth1 192.168.1.1 up
```

Appendix A - ADSL Protocol and PVC Values

Country	Provider	ATM Protocol	VPI	VCI
.au	Telstra, Internode, iiNet, TPG, Optus	PPPoE, PPPoA	8	35
.au	Nextep	Bridged Eth RFC1483.	1	32
Austria	AON	PPPoA	8	48
.be	Belgacom	PPPoE or Bridged Eth. RFC1483	8	35
.bh	Batelco	PPPoA	8	35
.es		Routed IP RFC1483	8	32
.ch	Sunrise and Bluewin	PPPoE	8	35
.fr	Wannadoo	PPPoA or PPPoE	8	35
.nl		PPPoA	8	48
.nz	NZ Telecom	PPPoA	0	100
.pt	Portugal Telecom	PPPoE or Bridged Eth. RFC1483	0	35
.uk	British Telecom	PPPoA	0	38
USA	Bell South	PPPoA	8	35
	Earthlink	PPPoE	0	35

Appendix B – CLI and ADSL Parameters

Included in the driver tarball on SourceForge is a simple CLI utility to allow access to the ADSL parameters for each port. Usage...

```
$soloscli -g Port Parameter
$soloscli -s Port Parameter Value
```

Notes :

g = get

s = set

Port is the ADSL port number with a value of 0 or 1

Parameter is one of the defined parameters from the list below.

Value is one of the defined Values associated with the parameter from the list below.

Parameter	Description	Action	Possible Values
FirmwareVersion	DSP Firmware Version.	GET	
Connected	It Indicates if DSL Line is in Data mode or not.	GET	True, False
OperationalMode	Indicates the ADSL mode the port has trained to.	GET	GET T1.413 g.lite G.Dmt Multimode ADSL2 ADSL2+ Inactive
State	Indicates the state of the DSL Line.	GET	Idle Showtime BitSwapping HandShake Training Discovery Training(DELT) DELT SpectrumTest
TxBitRate	Indicates the Line Tx Bit rate.	GET	
RxBitRate	Indicates the Line Rx Bit rate.	GET	
RxATTNDR	Maximum attainable rate Downstream	GET	
TxATTNDR	Maximum attainable rate Upstream	GET	
AnnexType	Indicates the AnnexType the line has trained to.	GET	AnnexA AnnexB AnnexM AnnexL Inactive
TxCellRate	Indicates the Cell Transmit Rate	GET	
RxCellRate	Indicates the Cell Receive Rate	GET	
PhyTXCellCount	Cells transmitted via PHY in TX direction	GET	
PhyRXCellCount	Cells received via PHY in RX direction	GET	
PhyCellDropCount	Indicates the PHY Cell Drop Count	GET	
RSCorrectedErrorsDn	Reed Solomon Corrected Errors count Downstream	GET	

Parameter	Description	Action	Possible Values
RSUnCorrectedErrorsDn	Reed Solomon UnCorrected Errors count Downstream	GET	
RSCorrectedErrorsUp	Reed Solomon Corrected Error count Upstream	GET	
RSUnCorrectedErrorsUp	Reed Solomon UnCorrected Error count Upstream	GET	
ShowtimeStart	Line up time in seconds.	GET	
ATURVendor	CPE Vendor Id String.	GET	CNXT
ATUCCountry	CO country code	GET	
ATURANSIRev	CPE ANSI Revision Number for T1.413	GET	
ATURANSISTD	CPE ANSI Standard/Version Number for T1.413	GET	
ATUCANSIRev	CO ANSI Revision Number for T1.413	GET	
ATUCANSIId	CO's ANSI Vendor Id.	GET	BRCM TXN ADI Aware ST/Alcatel IFX Ciena/Catena CTLM TexasIns AlcatelME AlcatelNtwrkS Cisco
ATUCANSISTD	CO ANSI Standard/Version Number for T1.413	GET	
LocalSEF	CPE SEF(Severely Errored Frames) count	GET	
LocalEndLOS	CPE LOS(Loss of Signal) count	GET	
LocalSNRMargin	CPE SNR Margin	GET	Returned in dB
LocalLineAttn	Downstream Line attenuation	GET	Returned in dB
INPup	Number of DMT symbols protected against impulse noise in Upstream	GET	
INPdown	Number of DMT symbols protected against impulse noise in Downstream	GET	
RawAttn		GET	Returned in dB
LocalTxPower	Upstream Transmit Power.	GET	Returned in dB
RemoteTxPower	Downstream Transmit Power	GET	Returned in dB
RemoteSEF	SEF count received by CO	GET	
RemoteLOS	LOS count received by CO	GET	
RemoteLineAttn	CO Rx Line Attenuation	GET	Returned in dB
RemoteSNRMargin	CO Rx SNR Margin	GET	Returned in dB
LineUpCount	Number of times the line has come up in SHOWTIME after system reboot.	GET	
Action	Startup	GET / SET	Startup, DELT, L3REQ, SpectrumReverb, SpectrumMedely, SpectrumPilot, MtsRequest

Parameter	Description	Action	Possible Values
ActivateLine	Indicates Start/Abort on the Line	GET / SET	None, Abort, Start
LineStatus	Flag indicating if the Line was given Abort/Start. It will be false if line is aborted, else true.	GET	True, False
Retrain	Option to configure the type of Retrain permissible on Line. Controls the Retrain conditions.	GET / SET	EnableAll, EnableOverallOnly, EnableCrcMinuteOnly, Disable, EnableOverallOnly
Status	Status of the DSL physical link	GET	Disabled, NoSignal, Initializing, EstablishingLink, Up
TotalStart	Number of seconds since the beginning of the period used for collection of Total statistics.	GET	

Appendix C – Firmware upgrades

C.1 ADSL Firmware Flash Upgrade

Copy the firmware binary image to:

```
/lib/firmware/solos-Firmware.bin
```

Reload the driver with the firmware upgrade module parameter set:

```
$ rmmod solos-pci.ko  
$ insmod solos-pci.ko firmware_upgrade=1
```

The modem status LEDs will turn off during the upgrade. When the upgrade is complete, the LEDs will return to their normal behaviour.

Status messages are written to `/var/log/messages` when the upgrade begins and ends. "Returning FPGA to Data mode" is written when the upgrade is complete.

The firmware upgrade should take between 2-3 minutes to complete.

APPENDIX D – ADSL Troubleshooting

D.1 Loading Modules

If the ADSL module loads without error messages, you should see the following in response to the **lsmod** command...

Module	Size	Used by
solos_pci	15751	1

You will probably also see several other kernel modules loaded.

You should also see the following in the system log (syslog command)...

Solos PCI Driver Version 0.07
solos 0000:00:0c.0: Solos FPGA Version 0.03 svn-38
solos 0000:00:0c.0: Registered ATM device 0
solos 0000:00:0c.0: Registered ATM device 1

D.2 Line Sync

If you have other devices sharing your ADSL line they must be connected via a filter or splitter to prevent them from interfering with your ADSL service. See the connection diagram in Section 2.3.

D.3 TX and RX packets

If you are using Bridged Ethernet or PPPoE you can use the **ifconfig** command to check if packets are being sent and received. If packets are being sent, but none are received check your ATM settings. Do you have the correct VPI and VCI for your provider? See Appendix A.

D.4 LCP timeouts (PPPoE and PPPoA only)

Check /var/log/messages for LCP timeouts. These are often caused by incorrect ATM settings. Check your VPI and VCI values against the table in Appendix A.

D.5 Authentication Failure (PPPoE and PPPoA only)

Check your username and password in the following files

/etc/ppp/chap-secrets

/etc/ppp/pap-secrets

Note : Some providers require an @provider in the username, eg.

mickey@bigpond.com

or

minnie@bt.co.uk

D.6 ADSL Troubleshooting

The ADSL line speed is affected by several factors...

- Line Length
- ADSL standard
- Signal to Noise Ratio (SNR)

You can check the line speeds via the CLI...

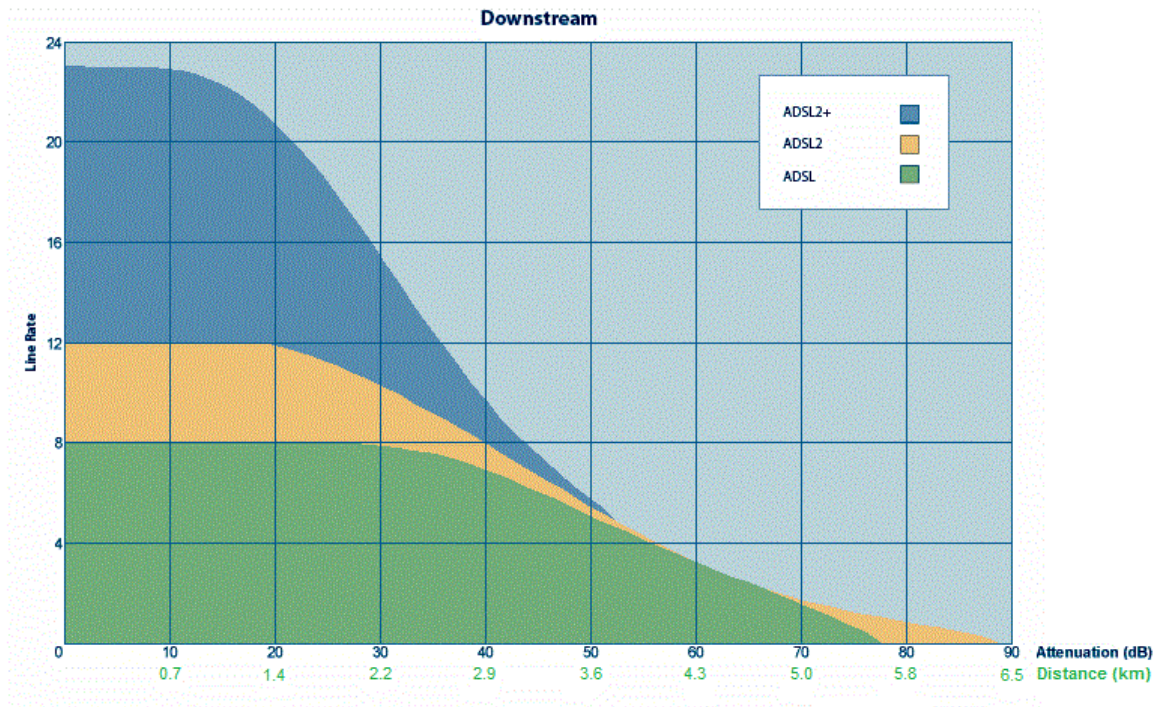
Linux# soloscli -g 0 RxBitRate {port 0}

or

Linux# soloscli -g 1 RxBitRate {port 1}

D.6.1 Line Length

As the graph below shows, speed is an inverse function of line length :



Important : the above graph is theoretical is does not take into account line condition, crosstalk or noise.

Shorter lines have less attenuation, especially at high frequencies, so there is more bandwidth for a given SNR. You can check the line attenuation via the CLI...

Linux# soloscli -g 0 LocalLineAttn {port 0}

or

Linux# soloscli -g 1 LocalLineAttn {port 1}

The result returned by the CLI is in dB, using the previous graph it is possible to estimate the line length.

D6.2 ADSL Standard

You can check the line standard via the CLI...

Linux# soloscli -g 0 OperationalMode {port 0}

or

Linux# soloscli -g 1 OperationalMode {port 1}

Possible values and maximum speeds are :

Standard	Max. Speed Downstream
T1.413	8Mbps
g.lite	2Mbps
G.Dmt	8Mbps
Multimode	-
ADSL2	12Mbps
ADSL2+	26Mbps
Inactive	0

For a given line length, the ADSL standard can have a large effect on line speed. The ADSL standard is configured in the DSLAM, and the Geos will automatically detect which standard to use. So for example if your ADSL line is configured for G.Dmt, then the maximum speed possible is 8Mbps. At longer line lengths this speed will drop according to the graph.

Also on very long or noisy lines the Geos may back off, I.e even though the line is configured for ADSL2+, training may fall back to ADSL2 or G.Dmt.

D6.3 Signal to Noise Ratio

The above graph is based on a Signal to Noise Ratio (SNR) of 6dB. 6dB is the lowest recommend SNR for good quality lines. Lower SNR values will produce faster speeds, but will be less stable. For poor quality or noisy lines an SNR of 10 or 12 dB may be required.

Normally the target SNR is configured in by the DSLAM. Your provider may allow you request a change of this (stability profile), or even provide a web interface to allow adjustment of the SNR.

You can check the SNR via the CLI...

Linux# soloscli -g 0 LocalSNRMargin {port 0}

or

Linux# soloscli -g 1 LocalSNRMargin {port 1}

In the situation where you need to increase the SNR (due to a poor quality or noisy line) it is possible to override the DSLAM configuration.

For ADSL2+ and a 10dB SNR...

Linux# soloscli -s 0 BisAMaxMargin Disable {default value is PerCO}

Linux# soloscli -s 0 BisAForceSNRMarginDn 0x60 {10dB}

Linux# soloscli -s 0 ActivateLine Start

For a 12dB or 14dB SNR use the following values instead in the above sequence...

soloscli -s 0 BisAForceSNRMarginDn 0x78 {12dB}

soloscli -s 0 BisAForceSNRMarginDn 0x8C {14dB}

APPENDIX E – Programmable LEDs & Switches

```
#GPIO 3 is the button SW1 near the USB socket
#GPIO 6 is LED0
#GPIO 25 is LED1
#GPIO 27 is LED2
```

```
modprobe cs5535_gpio major=222 mask=0x0E000048
for min in 3 6 25 27 ;
do
    mknod -m 0664 /dev/gpio_$min c 222 $min
done
```

```
#init the output LEDs
echo "Otdp" > /dev/gpio_6
echo "Otdp" > /dev/gpio_25
echo "Otdp" > /dev/gpio_27
```

```
#Turn LEDs on
echo "0" > /dev/gpio_6
echo "0" > /dev/gpio_25
echo "0" > /dev/gpio_27
```

```
#init the input
echo "IoTPd"> /dev/gpio_3
```

Then you should be able to read /dev/gpio_3
and test for the first character being a '0':

Button not pressed:

```
# cat /dev/gpio_3
1IoTPd
```

While button is pressed:

```
# cat /dev/gpio_3
0IoTPd
```

Copyright

Copyright © 2012

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Traverse Technologies Australia Pty. Ltd.

While reasonable efforts have been taken in the preparation of this manual to assure its accuracy, the manufacturer and distributors assume no liability resulting from any errors or omissions in this manual, or from the use of the information contained herein. The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.